# Enriched Internet Topology

Esteban Sergio Poggio[⋆1] and José Ignacio Alvarez-Hamelin[⋆⋆1,2]

[1] Facultad de Ingeniería UBA, Paseo Colón 850 C1063ACV–Buenos Aires, Argentina
[2] INTECIN (UBA–CONICET)
{esteban.poggio, ihameli}@cnet.fi.uba.ar

**Abstract.** We present a method to perform Internet tomography obtaining its enriched topology, i.e., adding to each link properties such as delay, utilization, etc. We consider *symmetrical* measurements, that is, we execute `traceroute`'s like probes using controllable nodes playing either source and target role. We based on PlanetLab testbed our experiments and we obtain an annotated map of a representative Internet portion. This information could be useful to build a thorough Internet model.

## 1 Introduction

This work aims to survey and to study the Internet topology, which is enriched with annotations such as bandwidth, delay, availability, statistical characteristics of traffic, loss rate, etc. Internet modelling arises from the necessity to have networks to test communication protocols, for example, to verify the operation of new routing algorithms in artificial networks having the same characteristics of real one. To improve existing models, further explorations are needed, such as adding to the topology more information like bandwidth or link utilization.

On the one hand, many works [1,2,3,4,5] study end-to-end properties such as available bandwidth, path capacity, loss rates and packet delays across an entire network, because it is often desirable to know the characteristics of the path which the application packets traverse through. On the other hand, there are several works [6,7,8,9] interested in building Internet topology at routers level as much as autonomous systems. Topology measurement studies consist of three phases: (i) topology collection, (ii) topology construction, and (iii) topology analysis. The map construction process includes an important step called IP alias resolution [10], the task of identifying IP addresses belonging to the same router in the collected data set. For instance, inaccuracies in alias resolution affects the representativeness of the resulting map.

For a network with $n$ end hosts, it is necessary to take $O(n^2)$ measurements and thus lack scalability due to the growth of $n$. In networks like Internet,

⋆ E.S.P. was partially supported by CONICET's fellowship.
⋆⋆ This work was supported by UBACyT-2010-20020090200119 and PICT-2010/1108. Authors also acknowledge to Artur Ziviani and Thiago Cardozo for helping us with FLAME platform.

there are many links which are shared between different paths, so the number of measurements is quite smaller than $O(n^2)$. Chen et al. [1,2] formulate this problem as follows: considering that a network with $n$ end hosts has $r = O(n^2)$ paths among them, they wish to select a minimal subset of $q$ paths to monitor so that the metric of all other paths can be inferred. Their algebraic model, which applies to any network topology, is described as follows: suppose a network spans $s$ links; they represent a path by a vector $v \in \{0,1\}^s$, where the $j$th entry $v_j$ is '1' if link $j$ is part of the path, and '0' otherwise. They form a rectangular matrix $G \in \{0,1\}^{r \times s}$ to represent $r$ paths. Then, they write the system of equations relating link measurements to path measurements as,

$$b = G \cdot x \ , \tag{1}$$

where $b$ are the measured values, $G$ is the routing matrix and $x$ is the metric (i.e., loss rate, delay, etc.). Normally, the number of $r$ paths is much larger than the number of $s$ links (see [2] Fig. 2(a)). This suggests that it could select $s$ paths to monitor, use those measurements to solve the linear system and infer the value of the other paths from (1). However, in general $G$ is rank deficient, specifically $q < s$, where $q = rank(G)$. In this case, it will be unable to determine the solution of some links from (1). These links are also called *unidentifiable* in network tomography literature [11]. Chen et al. [1,2] are not concerned about the characteristics of individual links, but the end-to-end properties. To select a basis set of $q$ paths to be monitored, the authors use standard rank-revealing decomposition techniques such as $QR$ decomposition [12]. The complexity for this step is $O(rq^2)$. Then, to compute the metric they must find a solution to the under-determined linear system. In this case, the complexity is $O(q^2)$. By extensively studying synthetic and real topologies, they found that for reasonably large $n$ (e.g., 100), $q$ is only in the range of $O(n \cdot log(n))$. Notice that the complexity is dominated by the QR decomposition with $O(n^4 \cdot \log^2(n))$.

Following the idea of how many paths to measure, then Chua et al. [3] recast the problem as one of statistical prediction and show that end-to-end network properties may be accurately predicted in many cases using significantly smaller set of carefully chosen paths than the needed ones for exact recovery. The authors analyse the routing matrices, and they observe that the rank $q$ of the routing matrix $G$ is an important quantity in regards to the sampling of paths for network monitoring. The relevance of this observation is the implication that networks with routing matrices $G$ of effective rank $q' \ll q$ may potentially allow for efficient monitoring using a very small number of paths (certain links in the network are more important to measure than others). To analyse the effective rank they use the SVD (*Singular Value Decomposition*, see [12]). They show a real case with Abilene network where they only measure $20-30\%$ of the paths. The complexity of this approach is dominated by E-BLP (*Estimated Best Linear Predictor*) on the Moore-Penrose generalized inverse matrix with $O(r^2 s) \approx O(n^5)$.

Shavitt et al. [5] introduce another approach which uses algebraic tools to compute distances that are not explicitly measured. One of the approaches to distance estimation in the Internet is based on placing Tracer stations in key

locations and conducting measurements between them. The Tracers construct an approximated map of the Internet after processing the information obtained from these measurements. The main idea behind their approach is that using the route measurements, one can identify nodes through which routes between several Tracers pass. They refer to these nodes as crossing points. In this way, they solve the following problem: given a set of end-to-end distances (i.e., delays) between Tracers with their associated routes, find all the possible segments[3] or groups of consecutive segments whose lengths can be derived. Their approach has several practical impacts. First, it can reduce the number of Tracers and measurements without sacrificing information. Second, their algorithm is able to compute distance estimates between locations where Tracers cannot be placed. Starting with $t$ Tracers one may reveal $n$ nodes. Using different hash tables one can identify the crossing points and the segments in $O(n)$ and writing the $n = O(t^2)$ equations in $O(ns)$, where $s$ is the number of segments. Triangulating the equations requires $O(nss') \approx O(n^3)$, where $s'$ is the number of solvable segments and checking which of the segments are solvable requires less than $O(ns^2) \approx O(n^3)$.

Finally, Allalouf et al. [4] present the design of a system for conducting large scale QoPC (*Quality of Path Characteristics*) measurements. The characteristics of the path which the application packets traverse through, are of great importance due to they determine the suitability of the path to the application requirements. In general, one way methods are preferred over round trip methods since they are less susceptible to measurement noise. The IDM (*Inter-packet Delay Measurement*) system was designed and developed to be a powerful tool for researchers, enabling them to perform one way measurements using packet trains emitters and sinks for determining the QoPC. The IDM system was designed as an extension to the DIMES [8] infrastructure. As the set of receivers, the authors select the ETOMIC [13] servers due to their ability to measure packet arrival time at sub $\mu$Sec accuracy.

## 2 The model

In this section we present the problem to be solved. We are concerned about the characteristics of individual links to build annotated Internet maps. Thus, we take measurements in both directions (we call it *symmetrical*). As a result we will obtain an overdetermined system, in other words, we will have more equations than unknowns. For this purpose we select 100 nodes and take measures all against all.

For the implementation of measurements we use a tool similar to `traceroute` called *trace*. It uses UDP datagrams trying to ensure, at least in outward trip, that all packets follow the same route, based on all packets sent which belong to the same data flow as used in [14]. In *per-flow load balancing*, packet header information ascribes each packet to a flow, and the routers forwards all packets

[3] A segment is a maximal sub-path of a measurement path, whose end-points are either Tracers or crossing points, that does not include an internal crossing point.

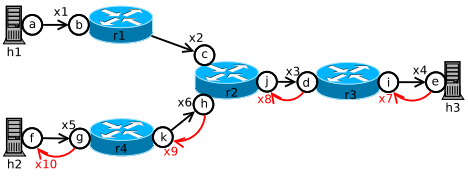



Fig. 1: A sample network: 3 end-hosts and 4 routers.

belonging to a same flow to the same interface. A natural flow identifier is the classic *five-tuple* of fields from the IP header and either the TCP or UDP headers: Source Address, Destination Address, Protocol, Source Port, and Destination Port. Path traces collected by this tool consist of IP addresses of devices which have a round trip time (RTT) and a time to live (TTL) values associated (i.e., $trace(h_i, h_j) = (T_i, ..., T_j)$), respectively, so each $T_i$ is composed of three values as follows: $T_i$ : $\{i$ (IP address), $TTL_i$, $t_i$ $(RTT_i)\}$. Then, we are interested in some specific characteristics such as minimum times and mean times. Below, we show how to build the routing matrix from measured data and we introduce a novel method to solve the system generated by the routing matrix.

*The Routing Matrix.* We assume that $r_i$ and $h_i$ identify routers and end hosts, respectively. The lower case letters $a, ..., k$ represent interface IP addresses, and each source has '0' TTL and '0' RTT. Figure 1 shows a known topology to illustrate the matrix construction method from measurement data $trace(h_i, h_j)$. Suppose that we have three path traces $trace(h_1, h_3) = (T_a, T_b, T_c, T_d, T_e)$, $trace(h_2, h_3)$ and $trace(h_3, h_2)$. For simplicity, we assume symmetric routing and we consider directed links. Similar to that developed in previous works [1,2,3,5] we can write the following system of linear equations:

$$t_b/2 = x_1 \ , \tag{2a}$$
$$t_c/2 = x_1 + x_2 \ , \tag{2b}$$
$$t_d/2 = x_1 + x_2 + x_3 \ , \tag{2c}$$
$$t_e/2 = x_1 + x_2 + x_3 + x_4 \ , \tag{2d}$$

where each $t_y/2$ represent the measured RTT (divided by two because of the symmetric routing assumption) at the interface $y$, and $x_i$ represents the metric to calculate for link $i$. Each of these equations define a part of the routing matrix $G$ in (1). Analysing $trace(h_2, h_3)$ and $trace(h_3, h_2)$ we notice the effect of directed links. Namely, for a link between $r_2$ and $r_3$ we have an unknown $x_3$ obtained from $trace(h_2, h_3)$, which is different to $x_8$ extracted from $trace(h_3, h_2)$. Then, doing the same for each $trace(h_i, h_j)$ and putting them together, we form a rectangular matrix $G$. Each row of $G$ represents a path in the network where $G_{i,j} = 1$ when path $i$ contains link $j$, and $G_{i,j} = 0$ otherwise. Sharing links between different paths is represented in the routing matrix as a new '1' in a column that already exists. For instance, based on topology depicted in Fig.1 we discover $x_3$ from $trace(h_1, h_3)$ and also from $trace(h_2, h_3)$, so $G_{i,3} = 1$ and $G_{j,3} = 1$, strictly

speaking, we will have a '1' in the same column and this is what generates the overdetermined system. Finally, we can solve the system defined in (1) applying LSQR [15] method, which can be used to solve a matrix which might be square or rectangular, overdetermined or under-determined, and might have any rank.

*Matrix Differences.* In general, the $G$ matrix is sparse; that is, there are only a few non-zeros per row, but it is possible to increase the number of zeros by subtracting equations that belong to the same output sequence, i.e., $trace(h_i, h_j)$. For instance, subtracting (2c) from (2b) we get,

$$\begin{aligned} t_d/2 - (t_c/2) &= x_1 + x_2 + x_3 - (x_1 + x_2) \ , \\ (t_d - t_c)/2 &= x_3 \ . \end{aligned} \tag{3a}$$

After applying the linear transformation we obtain an equivalent system:

$$b' = G' \cdot x \ , \tag{4}$$

where $G'$ is called *matrix differences*. Comparing the solution of the matrix $G'$ by the LSQR method with the solution of the matrix $G'$ averaging the measured values for a same column, we do not obtain any differences in the calculated results. Thus, we can remark that we have reduced the system complexity when solving the equivalent system. This fact reduces the time complexity, as it will be shown later.

*IP Alias Resolution.* Topology construction requires identification and grouping of IP addresses belonging to the same router. The goal of IP alias resolution is to identify the IP addresses that belong to the same router and combine them into a single node in the resulting topology map. To implement this task, we use a method known as APAR (*Analytical and Probed-based Alias Resolver*) [10]. APAR includes two steps: (1) analysing IP addresses in the set of collected path traces to identify a set of candidate subnets, and (2) using the identified subnets to resolve IP aliases. In our case we only consider point-to-point links, therefore we will use a /30 mask to identify subnets. Notice that taking *symmetrical* measurements help to solve IP alias problem because we have more possibilities to obtain useful information.

*Building $G'$.* After IP alias resolution we have identified the IP addresses that belong to the same router in the following data structure:

$$vertices = \{v_1 : [b, o], v_2 : [c, n], v_3 : [d, j, m], v_4 : [e], ..., v_8 : [a], ...\} \ , \tag{5}$$

where $v_i$ represents a vertex (in our case a router) and $[a, ..., i]$ represent a list of IP addresses that belong to that vertex. Iterating through $trace(h_i, h_j)$ and grouping consecutive pairs of interfaces that belong to the same output sequence, we can define two new data structures called *variables* and *measurements*, respectively. We must find to which vertex belongs each IP address obtained and

then concatenate these values to form a new variable $x_i$. This is exemplified from a sample network in Fig. 1. For example, analysing $trace(h_1, h_3)$ we observe:

$$\begin{cases} a \in v_8 \\ b \in v_1 \end{cases} \Longrightarrow v_8|v_1 = x_1 \Longrightarrow t(x_1) = [t_b/2] \ ,$$

$$\begin{cases} b \in v_1 \\ c \in v_2 \end{cases} \Longrightarrow v_2|v_1 = x_2 \Longrightarrow t(x_2) = [(t_c - t_b)/2] \ ,$$

where '|' denote concatenation.

Due to the fact that Internet is a best-effort network, routing packets through the same path is not guaranteed and packets travelling on the network may suffer different delay, reordering, etc., we observe negative times when we obtain the equivalent system defined in (4). We take advantage of *symmetrical* measurements and we apply one of the following solutions:

1. **Reverse link**: we have considered directed links, so we have information about both directions. If we observe a positive time in the opposite direction, we will replace this measurement by this value. Thus, this link becomes undirected.
2. **Virtual link**: this concept is defined in [1]. Given that a negative time is useless for our approach, the metric of these links is not possible to calculate, so these ones are *unidentifiable* links. If it is not possible to use **Reverse link** solution, we will define a virtual link searching for a positive time towards successor or predecessor vertices, that is, this virtual link is covered by some path segment whose metric is identifiable.

As a result of iterating data structure called *vertices* we will obtain:

$$variables = \{v_1|v_8 : x_1, v_2|v_1 : x_2, \ \ldots, v_j|v_i : x_n\} \ , \tag{6}$$

where $v_i|v_j$ denotes a pair of joined vertices and $x_k$ denotes the metric for that link, and

$$\begin{aligned} measurements = \{&t(x_1) : [t_1(x_1), t_2(x_1), \ldots, t_x(x_1)], \\ &\vdots \\ &t(x_r) : [t_1(x_r), t_2(x_r), \ldots, t_z(x_r)]\} \ , \end{aligned} \tag{7}$$

where $[t_1(x_i), t_2(x_i), \ldots, t_n(x_i)]$ is the vector of measurements associated with the unknown $x_i$.

From the *measurements* data structure we can solve directly the system defined in (4). Namely, if the vector of values associated with $t(x_i)$ has more than one component, we will average all values to obtain the solution of unknown $x_i$; and if the vector of values associated with $t(x_i)$ has only one component, it will be the solution of unknown $x_i$. Once we have solved the system, from the *variables* data structure we can construct the graph in a standard format which in general we have three values per row: $v_j \ v_i \ x_n$. Then, we analyse the graph obtained with `Network Workbench`[4]. Notice that we will build a graph for each analysis case, that is, one for mean times and another one for minimum times.

[4] nwb.cns.iu.edu

*Complexity.* Let $s$, $d$ and $e$ denote the number of sources, targets and experiments, respectively; we define the complexity of the whole problem as $O(n)$, where $n = s\,d\,e$. Then, we extract information from measurements $trace(h_i, h_j)$ in $O(n)$ and we identify subnets in $O(n)$ too. Making the data structure defined in (5) has the highest complexity, strictly speaking, we can solve IP alias problem in $O(n\,e^2) \approx O(n^3)$. Making the data structures defined in (6) and (7) require both $O(n)$. Finally, solving the system defined in (4) and building the graph require both $O(n)$.

## 3 Experimental results

In this section we describe briefly how the measurements have been developed and the analysis of the experimental results.

*Methodology.* PlanetLab[5] is a global research network that supports the development of new network services. We used this testbed for our measurement study and we deployed it on 100 PlanetLab hosts which belongs to different sites.

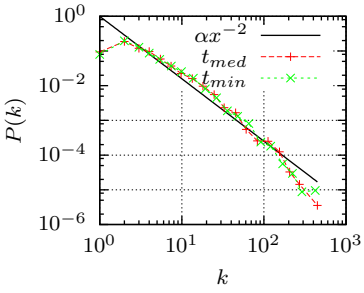
Through FLAME (*Flexible Lightweight Active Measurement Environment*) [16] platform we measure some network characteristics between PlanetLab nodes. FLAME is based on the distribution of measurement agents among some network nodes. Such agent send and receive probe packets in response to commands from a central manager. Users issue such commands to the central manager with a command line-based console. The agents return the collected measurement data to the central manager, which publishes such data in a standardized way on a central repository, simplifying the management and further analysis of such data. All communication among the three components is based on the XMPP [17] protocol. Using FLAME we implement a similar tool to well-known `traceroute`, which we called *trace* due to the similarity to the one in [10].

The aim of exploration is to achieve a large-scale deployment during a long period of time. It seeks to make observations between 100 nodes (all against all) for 24 hours, where the probe packet is executed on PlanetLab agents 5 times per hour, resulting in a total of 120 measurements per node. First, when we have tried to achieve this goal, we have found a limitation in FLAME platform due to a blocking function that does not allow to follow running a single probe up to obtain a response. Finally, we have avoided this problem dividing the original slice in five different slices. Thus, we have been able to fulfil the proposed task.

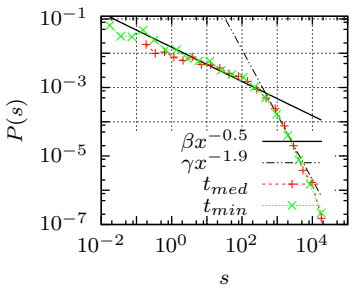
*Results.* The topological properties of a graph are fully encoded in its weighted adjacency matrix $W$, whose entry $w_{ij}$ gives the weight on the edge connecting the vertices $i$ and $j$, and '0' otherwise (for unweighted graphs $w_{ij} = a_{ij} = 1$). The indices $i$, $j$ run from 1 to $n$, where $n$ is the size of the network.

The degree distribution $P(k)$ of a network is defined to be the fraction of nodes in the network with degree $k$ (i.e., considering the graph as an unweighted one). Figure 2a illustrates the degree distribution $P(k)$ for two cases of analysis proposed. This result is similar to other Internet explorations [6,8].

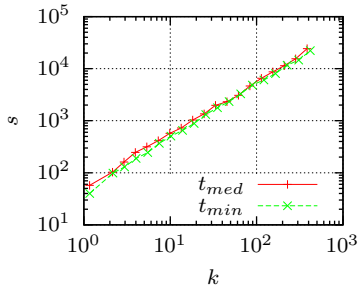
[5] www.planet-lab.org




(a) Degree distribution




(b) Strength distribution



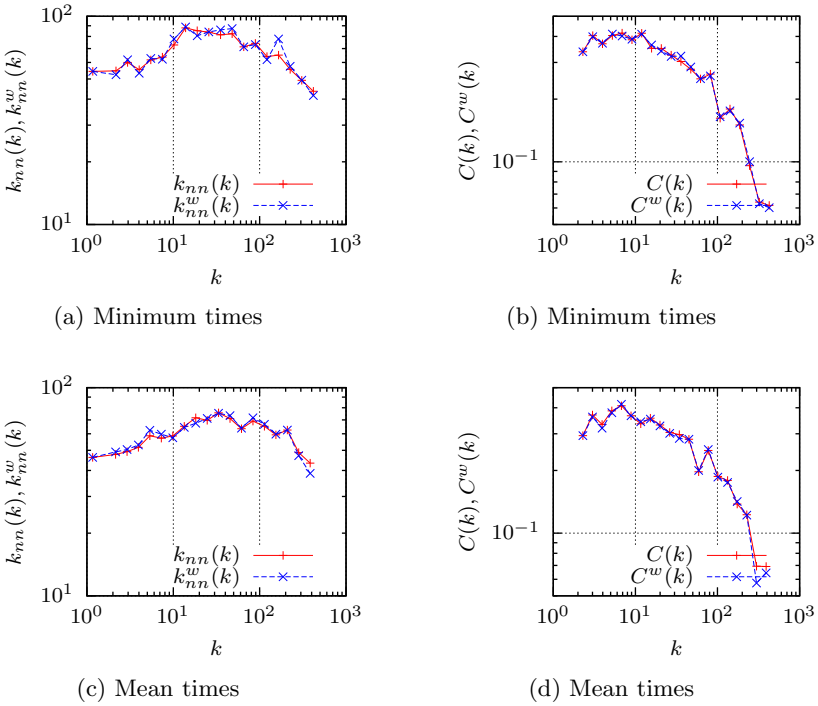

(c) Strength vs. grado

Fig. 2: Degree and strength distributions

A natural generalization in the case of weighted networks is the strength $s_i$. The strength of a node combines the information about its connectivity and the intensity of the weights of its links. Due to strength $s$ does not have a discrete value, we define intervals in a log-binning way. Several works [18,19,20] have shown that $P(s)$ corresponds to a distribution with heavy tail bounded by a power-law. Figure 2b illustrates the strength distribution $P(s)$ for two cases of analysis proposed (with a log-binning). Notice that curves follow a power-law distribution with a power-law cut-off (close to $\gamma = -1.9$), probably due to finite size of the obtained network. At last, Fig. 2c shows the relationship between strength and degree of the vertices for two cases of analysis, showing a strong correlation between both parameters.

Another important source of information lies in the correlations of the degree of neighbouring vertices. The weighted average neighbour degree distribution can be deduced from [21] as,

$$k_{nn}^{w}(k) = \frac{1}{n \cdot P(k)} \sum_{\forall i:k_i=k} \frac{1}{s_i} \sum_{j=1}^{n} w_{ij} k_j \ , \tag{8}$$




(a) Minimum times (b) Minimum times

(c) Mean times (d) Mean times

Fig. 3: Graph parameters

where the first sum is the average of all $i$ vertices with degree $k$, and $n \cdot P(k)$ gives the number of vertices of degree $k$. For the unweighted case we have to replace $w_{ij}$ and $s_i$ by $a_{ij} = 1$ and $k_i$, respectively.

Figure 3a and Fig. 3c show the average neighbour degree distribution $k^w_{nn}(k)$, each one for weighted and unweighted cases, for minimum and mean times, respectively. We observe that both curves have almost no slope, so we could say, if we choose a random node in the unweighted case, the degree of its neighbours will not depend on its own degree. This same behaviour is observed in the weighted case, due to the strong correlation between degree and strength, as it is shown in Fig. 2c.

The clustering coefficient is a measure of degree to which nodes in a graph tend to cluster together. The weighted clustering coefficient of a vertex $i$ is defined as [21],

$$c^w_i = \frac{1}{s_i(k_i - 1)} \sum_{j,h} \frac{(w_{ij} + w_{ih})}{2} a_{ij} a_{ih} a_{jh} \ , \tag{9}$$

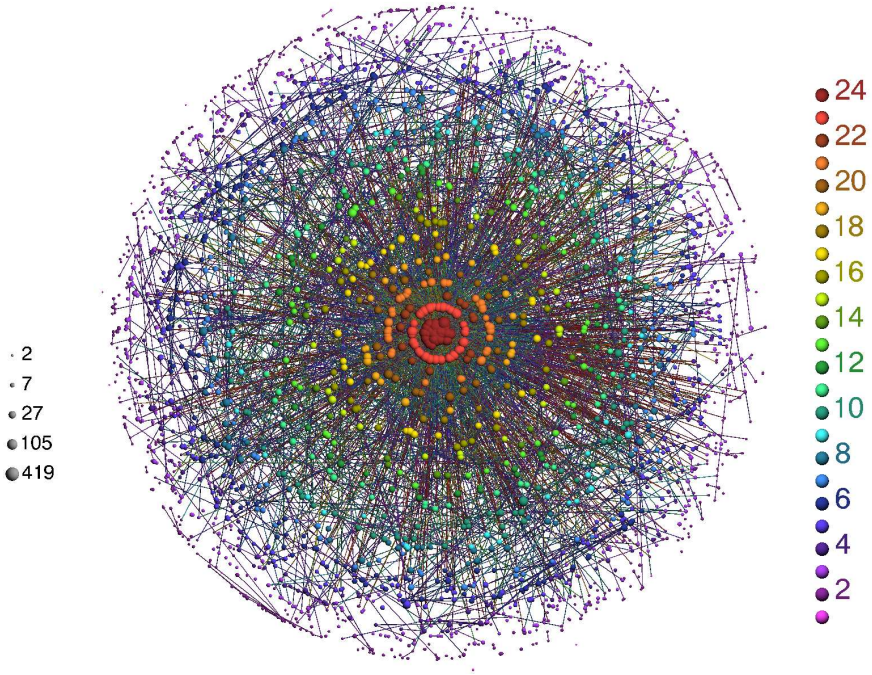



Fig. 4: Graph visualization of the topology obtained for the $t_{min}$ case.

if we consider unweighted graphs, we will obtain $c_i$ where $(w_{ij} + w_{ih})/2 = 1$ and $s_i$ is replaced by $k_i$. The clustering coefficient distribution as a $k$ function indicates what probability have the neighbours of a vertex with $k$ degree to be interconnected, and can be expressed as [21],

$$C(k) = \frac{1}{n \cdot P(k)} \sum_{i/k_i=k} c_i \,. \tag{10}$$

Figure 3b and Fig. 3d show the clustering coefficient distribution $C(k)$ and $C^w(k)$ for minimum and mean times, respectively. The similarity between the curves $C(k)$ and $C^w(k)$ is explained on the basis of the aforementioned relation between the strength and degree (Fig. 2c).

Finally, for graph visualization we use the LaNet-vi (*Large Networks visualization*)[6] tool that is based on $k$-core decomposition [22]. Figure 4 illustrates the graph visualization of the topology obtained for the $t_{min}$ case. On the right it shows a color scale with the shell-index number, while on the left it shows in a logarithmic scale the vertices' degree represented by the size of them. Although there are some connections to the central core, there are also features of the router maps. There is a trend present where all layers are densely populated

[6] http://lanet-vi.soic.indiana.edu/

and connections (edges) occur mainly between layers. On the periphery we can observe nodes with a substantial size, where the size of these nodes is slightly reduced compared to those found in other studies, i.e., [23], this is due to the limited number of sources from which measurements were made.

## 4 Conclusions

In this work we achieve Internet tomography using PlanetLab and taking *symmetrical* measurements. This symmetrical way considers nodes as sources and targets, which make easier IP alias resolution. Later, with stored data and appropriate processing, we build annotated Internet maps for two parameters of interest: minimum and mean times. These parameters give us an idea of different network characteristics. On the one hand, the minimum time is an indicator of the physical length of the links. On the other hand, the mean time is an indicator of link congestion. Finally, and most importantly, the proposed processing allows working with large networks because of its low complexity.

Although *a priori* we expected to observe a marked difference between the average neighbour degree and the clustering coefficient for weighted and unweighted graphs, this did not happen. The justification lies in the strong correlation between degree and strength distributions. Moreover, it is remarkable that there are not any differences comparing the $t_{min}$ with the $t_{med}$ case for both distributions (weighted clustering coefficient and weighted average neighbour degree). Even though the model that we consider for the analysis of the mean times is a simple one, we might think that each link is independent of another, suggesting a proper network operation. For instance, if a link is saturated, it will not influence the congestion of other neighbour links.

In the future there will be open several possibilities to continue studying this issue. Therefore, it could be possible to increase the number of measurements executing `traceroute`'s like probes towards non-controllable nodes (i.e., *asymmetrical*), this means that even though we could lose the possibility of obtaining characterization in both link directions, we might get a more representative map based on undirected measurements.

## References

1. Chen, Y., Bindel, D., Katz, R.H.: Tomography-based overlay network monitoring. In: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement. IMC '03, New York, NY, USA, ACM (2003) 216–231
2. Chen, Y., Bindel, D., Song, H.H., Katz, R.H.: An algebraic approach to practical and scalable overlay network monitoring. In: SIGCOMM. (2004) 55–66
3. Chua, D.B., Kolaczyk, E.D., Crovella, M.: Efficient monitoring of end-to-end network properties. In: In Proc. of IEEE INFOCOM. (2005) 1701–1711
4. Allalouf, M., Kaplan, E., Shavitt, Y.: On the feasibility of a large scale distributed testbed for measuring quality of path characteristics in the internet (2009)

5. Shavitt, Y., Sun, X., Wool, A., Yener, B.: Computing the unmeasured: An algebraic approach to internet mapping. In: in IEEE INFOCOM. (2001)
6. McRobb, D., Claffy, K., Monk, T.: Skitter: Caida's macroscopic internet topology discovery and tracking tool. Available from http://www.caida.org/tools/skitter/ (1999)
7. Spring, N., Wetherall, D., Anderson, T.: Scriptroute: A Public Internet Measurement Facility (2002)
8. Shavitt, Y., Shir, E.: Dimes: let the internet measure itself. Computer Communication Review **35** (2005) 71–74
9. Paxson, V., Mahdavi, J., Adams, A., Mathis, M.: An architecture for large scale Internet measurement. IEEE Communications Magazine **36**(8) (August 1998) 48–54
10. Gunes, M.H., Sarac, K.: Resolving IP aliases in building traceroute-based internet maps. IEEE/ACM Trans. Netw. **17**(6) (December 2009) 1738–1751
11. Bu, T., Duffield, N., Presti, F.L.: Network tomography on general topologies. In: in ACM SIGMETRICS. (2002) 21–30
12. Golub, G.H., Van Loan, C.F.: Matrix computations (3rd ed.). Johns Hopkins University Press, Baltimore, MD, USA (1996)
13. Morato, D., Magana, E., Izal, M., Aracil, J., Naranjo, F., Astiz, F., Alonso, U., Csabai, I., Haga, P., Simon, G., Steger, J., Vattay, G.: The european traffic observatory measurement infraestructure (etomic): A testbed for universal active and passive measurements. Testbeds and Research Infrastructures for the Development of Networks & Communities, International Conference on **0** (2005) 283–289
14. Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., Teixeira, R.: Avoiding traceroute anomalies with paris traceroute. In: In Proc. Internet Measurement Conference. (2006)
15. Paige, C.C., Saunders, M.A.: Lsqr: An algorithm for sparse linear equations and sparse least squares. ACM Trans. Math. Softw. **8**(1) (March 1982) 43–71
16. Ziviani, A., Gomes, A.T.A., Kirszenblatt, M.L., Cardozo, T.B.: Flame: Flexible lightweight active measurement environment. In: TRIDENTCOM. (2010) 526–541
17. Saint-Andre, P.: Extensible Messaging and Presence Protocol (XMPP): Core (October 2004)
18. Guimera, R., Mossa, S., Turtschi, A., Nez-Amaral, L.A.: Structure and efficiency of the world-wide airport network. Preprint 0312535, available from http://arxiv.org/abs/cond-mat/ (2003)
19. Barrat, A., Barthelemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. PROC.NATL.ACAD.SCI.USA **101** (2004) 3747
20. Garlaschelli, D., Battiston, S., Castri, M., Servedio, V.D.P., Caldarelli, G.: The scale-free topology of market investments (2003)
21. Barrat, A., Barthélemy, M., Vespignani, A.: Modeling the evolution of weighted networks. Phys. Rev. E **70** (2004) 066149
22. Beiró, M.G., Alvarez-Hamelin, J.I., Busch, J.R.: A low complexity visualization tool that helps to perform complex systems analysis. New Journal of Physics **10**(12) (2008) 125003
23. Alvarez-Hamelin, J.I.: Taxonomía de los Modelos de Topología de Internet. In: Mecánica Computacional, Interdisciplinary Mathematical Methods. Volume XXV., CIMEC (2006) 2597–2612